Convex Feasibility Methods for Compressed Sensing

Avishy Carmi and Pini Gurfil, *Member, IEEE*

Manuscript received ; revised .

A. Carmi is with the Asher Space Research Institute, Technion – Israel Institute of Technology, Haifa 32000, Israel.

P. Gurfil is with the Faculty of Aerospace Engineering, Technion – Israel Institute of Technology, Haifa 32000, Israel.

Abstract

We present a computationally-efficient method for recovering sparse signals from a series of noisy observations, known as the problem of compressed sensing (CS). CS theory requires solving a convex constrained minimization problem. We propose to transform this optimization problem into a convex feasibility problem (CFP), and solve it using subgradient projection methods, which are iterative, fast, robust and convergent schemes for solving CFPs. As opposed to some of the recently-introduced CS algorithms, such as Bayesian CS and gradient projections for sparse reconstruction, which become prohibitively inefficient as the problem dimension and sparseness degree increase, the newly-proposed methods exhibit a marked robustness with respect to these factors. This renders the subgradient projection methods highly viable for large-scale compressible scenarios.

I. Introduction

Recent studies have shown that sparse signals can be recovered accurately using less observations than predicted by the Nyquist/Shannon sampling principle; the resulting theory is known as compressed sensing (CS) [1], [2]. The essence of the new theory builds upon a new data acquisition formalism, in which compression plays a fundamental role. Sparse – and more generally – compressible signals arise naturally in many fields of science and engineering, such as the reconstruction of images from under-sampled Fourier data, biomedical imaging and astronomy [3], [4]. Other applications include model-reduction methods to enforce sparseness for preventing over-fitting and for reducing computational complexity and storage capacities. The reader is referred to Refs. [1] and [2] for an extensive overview of CS theory.

The recovery of sparse signals consists of solving an NP-hard minimization problem [1], [5]. State-of-the-art methods for addressing this optimization problem commonly utilize convex relaxations, non-convex local optimization and greedy search mechanisms. Convex relaxations are used in various methods such as least absolute shrinkage and selection operator (LASSO) [6], least angle regression (LARS) [7], the Dantzig selector (DS) [8], basis pursuit (BP) and basis pursuit de-noising [9]. The two former schemes, the LASSO and LARS, are essentially homotopy methods that exploit pivoting operations on sub-matrices from the complete sensing matrix (sub-sensing matrices) for yielding a solution path to the convex optimization problem. These methods turn out to be extremely efficient whenever the sparseness level is relatively high owing to the fact that only a few sub-sensing matrices need to be provided, corresponding to the instantaneous support of the underlying reconstructed signal. The latter methods, the DS and the BP variants, recast a linear program and employ either simplex or interior-point techniques for obtaining an optimal solution. Similarly to the homotopy-based approaches, these methods may become computationally-intensive, since the number of elements in the support of the underlying signal increases.

Non-convex optimization approaches include Bayesian methodologies such as the relevance vector machine (RVM), otherwise known as sparse Bayesian learning [10] as well as stochastic search algorithms, which are mainly based on Markov chain Monte Carlo (MCMC) techniques [11]–[14]. In virtue of their Bayesian mechanism and in contrast to other optimization approaches, these methods provide a complete statistical solution to the problem by means of a probability density function. Nevertheless, the intensive computational requirements of these methods render their applicability questionable in high-dimensional problems.

Recently, the Bayesian framework was utilized to create efficient CS schemes [15], [16]. The Bayesian CS algorithm in [15] exploits both a sparseness-promoting hierarchical prior and an RVM mechanism for deriving point estimates and statistical error bounds. This method was shown to outperform some of the commonly-used greedy schemes both in accuracy and speed. The work in [16] derived a pseudo-measurement-based Kalman filtering algorithm (CSKF) for recovering sparse signals from noisy observations in a sequential manner. This approach extends CS to accommodate stochastic linear filtering problems and, similarly to the aforementioned Bayesian methods, yields a complete statistical solution to the problem (a Gaussian distribution).

Notable greedy search algorithms are the matching pursuit (MP) [17], the orthogonal MP [18], and the orthogonal least squares (OLS) [19]. Both MP and OMP minimize the reconstruction error by iteratively choosing elements from the dictionary matrix (sensing matrix). The OMP involves an additional orthogonalization stage and is known to outperform the conventional MP. The OLS works in a similar fashion, employing an orthogonal transformation of the original sensing matrix. The greedy algorithms are known to posses an advantage in terms of computational cost over other optimization schemes when the basis projections are sufficiently incoherent and the number of elements in the support is relatively small. This, in turn, implies that their performance may deteriorate dramatically in common realistic settings wherein the underlying signals are compressible rather than sparse.

The scalability of a CS technique essentially refers to its viability for high dimensional settings. This issue is given a prime consideration in [20], where a new gradient projection (GP)-based method is proposed for solving possibly large-scale CS problems. As pointed out in [20], large-scale methods require only matrix-vector products involving the sensing matrix. As demonstrated in [20], the GP algorithm is efficient in high-dimensional settings compared to the aforementioned methods. However, similarly to the homotopy methods, the practical implementation of the GP algorithm requires a delicate tuning of the l_1-norm bounding parameter, which greatly affects its convergence and optimality.

A. *Restricted Isometry Property and Convex Relaxations*

The theory of compressed sensing has drawn much attention to the convex relaxation methods. It has been shown that the convex l_1 relaxation yields an exact solution to the recovery problem provided two conditions are met: 1) the signal is sufficiently sparse, and 2) the sensing matrix obeys the so-called *restricted isometry property* (RIP) at a certain level. Another complementary result ensures high accuracy when dealing with noisy observations. Further elaboration of this result facilitated its probabilistic version, which is concluded by the known statement of recovery 'with overwhelming probability' [1]. To put it informally, it is highly probable for the convex l_1 relaxation to yield an exact solution provided the involved quantities, the sparseness degree s, and the sensing matrix dimensions $m \times n$ maintain a relation of the type $s = \mathcal{O}(m/\log(n/m))$.

B. *Convex Feasibility Problems, Subgradient Projections, and Compressed Sensing*

The issue of computational efficiency is of prime importance for large-scale CS problems. Ideally, a computational scheme for recovering sparse and/or compressible signals should be characterized by (i) computational efficiency; (ii) minimum mean-square error of the estimated signals; and (iii) scalability. Additional desirable features would be simple and transplant implementation (no "black boxes"); robustness with respect to some sparsensess/compressibility measure; and ubiquitous applicability to real-world problems.

A potentially promising approach for satisfying the aforementioned goals is to transform the CS problem into a convex feasibility problem (CFP). CFPs are aimed at finding a point in the intersection set of nonlinear convex sets. The CFP is a fundamental problem in numerous fields, see, e. g., [21], [22] and references therein. It has been used to model real-world problems in image reconstruction from projections [23], in radiation therapy treatment planning [24], and in crystallography [25].

Most often, CFPs are solved by performing projections onto the individual sets. This is carried out in various ways, by using different projection methods, resulting in a myriad of algorithms that usually exhibit good convergence while consuming minimal computational resources. Some of the projection algorithms may even provide predictable performance when the solution set of the CFP is empty [26]. We refer the reader to Ref. [27] for a discussion on the connection between projection methods and variational inequalities; to Refs. [28], [29] for applications in signal processing; and to [30, Chapter 5] for a comprehensive review.

However, projections onto convex sets may be prohibitively difficult when the constraints are nonlinear. This is so because computing orthogonal projections onto arbitrary convex sets requires a separate, inner-loop minimization for finding the minimum distance between a given point and an arbitrary curve, a process that usually involves a considerable computational effort.

An elegant alternative is to use *subgradient projections*. Subgradient projections only require the instantaneous subgradient in order to perform the next iteration. The projection is performed onto an intermediate point, and not directly onto the convex set. Some of the subgradient projection methods, such as cyclic subgradient projections [31] have a convergence proof when the intersection set of all the convex constraints is non-empty.

In this paper, we use both cyclic and simultaneous subgradient projections (CSP and SSP) for efficiently solving the CS problem. The proposed algorithms are characterized by (i) computational efficiency; (ii) accuracy; (iii) robustness to varying compressibility and sparseness levels; and (iv) straightforward, transparent implementation. By using extensive numerical evaluations, we illustrate the superiority of the proposed scheme – in terms of computational efficiency and accuracy – compared to commonly-used methods for solving the CS problem.

II. Sparse Signal Recovery and Compressed Sensing

Consider a signal $\chi \in \mathbb{R}^n$ that is sparse in some domain, i.e., it can be represented using a relatively small number of projections in some known, possibly orthonormal, basis, say $\psi \in \mathbb{R}^{n \times n}$. Thus, we may write

$$\chi = \psi x = \sum_{i=1}^{n} x_i \psi_i = \sum_{x_j \in \mathrm{supp}(x)} x_j \psi_j, \qquad \|x\|_0 < n \tag{1}$$

where $\mathrm{supp}(x)$ and $\|x\|_0$ (the l_0 norm) denote the support of x and its dimension (i. e., the number of non-zero elements of x), respectively. The problem of compressed sensing considers the recovery of x (and therefore of χ) from a limited number, $m < n$, of incoherent and possibly noisy measurements (or, in other words, sensing a compressible signal from a limited number of incoherent measurement) [1]. The measurements themselves obey a linear relation of the form

$$y = H' \chi = Hx \tag{2}$$

where $H \in \mathbb{R}^{m \times n}$ and $H = H' \psi$. In many practical applications, the observation vector y may be either inaccurate or contaminated by noise. In this case, which will be referred to as the *stochastic CS problem*, an additional noise term is added to the right-hand side of (2).

In general, under certain limitations on the sparseness degree of x, $s = \|x\|_0$, an exact solution to the above recovery problem can be obtained by essentially solving a subset selection problem of the form

$$\min_{\hat{x}} \|\hat{x}\|_0 \text{ s.t. } \|y - H\hat{x}\|_2^2 \leq \epsilon \tag{3}$$

for a sufficiently small ϵ. However, the problem (3) is known to be NP-hard, which implies that in practice an optimizer $\hat{x}$ cannot be computed efficiently.

In the late 90's, the l_1 norm was suggested as a sparseness-promoting term in the seminal work introducing the acclaimed LASSO operator [6] and the basis pursuit [9]. Recasting the sparse recovery problem using the l_1 norm yields a convex relaxation of the original NP-hard problem, which can be efficiently solved via a myriad of well-established optimization techniques. Commonly, there are two equivalent convex formulations that follow from (3): The quadratically-constrained linear program, which takes the form

$$\min_{\hat{x}} \|\hat{x}\|_1 \text{ s.t. } \|y - H\hat{x}\|_2^2 \leq \epsilon \tag{4}$$

and the quadratic program

$$\min_{\hat{x}} \|y - H\hat{x}\|_2^2 \text{ s.t. } \|\hat{x}\|_1 \leq \epsilon' \tag{5}$$

It can be shown that for proper values of the tuning parameters ϵ and ϵ' the solution of both these problems coincide.

Recently, [1], [2] have shown that an accurate solution of (3) can almost always be obtained by solving the convex relaxation (4) assuming that the sensing matrix H obeys the RIP mentioned before. The RIP roughly implies that the columns of a given matrix nearly behave like an orthonormal basis. This desired property is possessed by several random constructions, which guarantee the uniqueness of the sparse solution. In particular, an exact recovery is highly probable when using such matrices provided that a relation of the type

$$s = \mathcal{O}(m/\log(n/m)) \tag{6}$$

holds. For an extensive overview of several RIP constructions and their role in CS, the reader is referred to [1], [2].

In practice, the projected signal x may be *nearly* sparse, in the sense of having many relatively small elements, which are not identically zero. Such representations, frequently encountered in real-world applications, are termed *compressible*. Most of the results in the CS literature naturally extend to the compressible case assuming some behavior of the small nonzero elements. Such a behavior is suggested in [2], where the compressible element sequence is assumed to decay according to the power law

$$|x_i| \leq \kappa i^{-1/r}, \qquad |x_i| \geq |x_{i+1}| \tag{7}$$

where $\kappa > 0$ and $r > 0$ are the radius of a weak l_r ball to which x is confined, and a decay factor, respectively. In this case, an equivalent measure of the signal sparseness degree, s, can be obtained as

$$\hat{s} = n - \text{card}\{i \mid |x_i| \leq \varepsilon\} \tag{8}$$

for some sufficiently small $\varepsilon > 0$, where $\text{card}\{\cdot\}$ denotes the cardinality of a set.

III. Subgradient Projections for Compressed Sensing

In this section, we outline the convex feasibility problem and two algorithms for a solution thereof: The cyclic subgradient projections (CSP) and the simultaneous subgradient projections (SSP). We will explain how these algorithms and their variants, collectively referred to as *convex feasibility methods*, are implemented for efficiently solving the convex CS problem described by Eqs. (4) or (5).

A. The Convex Feasibility Problem

Given p closed convex subsets $Q_1, Q_2, \cdots, Q_p \subseteq \mathbb{R}^n$ of the n-dimensional Euclidean space, expressed as

$$Q_i = \{z \in \mathbb{R}^n \mid f_i(z) \leq 0\}\,, \tag{9}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$ is a convex function, the *convex feasibility problem* (CFP) is

$$\text{find a point } z^* \in Q := \cap_{i=1}^p Q_i. \tag{10}$$

If $Q \neq \emptyset$, the CFP is said to be *consistent*. Consequently, it is required to solve the system of convex inequalities

$$f_i(z) \leq 0, \quad i = 1, 2, \ldots, p. \tag{11}$$

The context of convex inequalities gives rise to the realm of subdifferential calculus, in which the definitions of subgradients and subdifferentials play a fundamental role. Given the convex function $f_i : \mathbb{R}^n \to \mathbb{R}$, a vector $t \in \mathbb{R}^n$ is called a *subgradient* of f_i at point z_0 if

$$f_i(z) - f_i(z_0) \geq \langle t, z - z_0 \rangle \tag{12}$$

The subdifferential of f_i at z_0, denoted by $\partial f_i(z_0)$, is the non-empty compact convex set

$$\partial f_i(z_0) := \{t : f_i(z) - f_i(z_0) \geq \langle t, z - z_0 \rangle, \forall z\} \tag{13}$$

When f_i is differntiable at z_0, the subgradient becomes a gradient, $t = \nabla f_i(z_0)$.

B. Subgradient Projections

Subgradient projections have been incorporated in iterative algorithms for the solution of CFPs. They can be roughly categorized into two main categories, *sequential* and *simultaneous*. The cyclic subgradient projections (CSP) method for the CFP, which is a sequential subgradient projections algorithm, was developed by Censor and Lent [31] and is summarized in Algorithm 1.

A convergence result for the CSP method in the consistent case was provided in [30, Chapter 5], where it was shown that if the functions $f_i(z)$ are continuous and convex on $\mathbb{R}^n$ $\forall i$; $Q := \cap_{i=1}^p Q_i \neq \emptyset$; and the subgradient is uniformly bounded, then any sequence $\{z^k\}$ produced by Algorithm 1 converges to a solution of the CFP, i. e., $z^k \to z^*$ as $k \to \infty$. The convergence proof in [30, Chapter 5] is based

Algorithm 1 The method of cyclic subgradient projections (CSP)

Initialization: $z^0 \in \mathbb{R}^n$ is arbitrary.

Iterative step: Given z^k, calculate the next iterate z^{k+1} by

$$z^{k+1} = \begin{cases} z^k - \alpha^k \dfrac{f_{i(k)}(z^k)}{\parallel t^k \parallel_2^2} t^k, & \text{if } f_{i(k)}(z^k) > 0, \\ z^k, & \text{if } f_{i(k)}(z^k) \leq 0, \end{cases} \tag{14}$$

where $t^k \in \partial f_{i(k)}(z^k)$ is a subgradient of $f_{i(k)}$ at the point z^k, and the relaxation parameters $\{\alpha^k\}_{k=0}^{\infty}$ are confined to an interval $\epsilon_1 \leq \alpha^k \leq 2 - \epsilon_2$, for all $k \geq 0$, with some, arbitrarily small, $\epsilon_1, \epsilon_2 > 0$.

Constraint-Index Control: The sequence $\{i(k)\}_{k=0}^{\infty}$ is cyclic, that is, $i(k) = k(\mod p) + 1$ for all $k \geq 0$.

on the concept of Fejér monotonicity: A sequence $\{z^k\}_{k=0}^{\infty}$ is *Fejér monotone* with respect to some fixed set $Q \subseteq \mathbb{R}^n$, if $\forall z \in Q$,

$$\|z^{k+1} - z\|_2 \leq \|z^k - z\|_2, \forall k \geq 0 \tag{15}$$

Sequential projection methods for solving CFPs usually have simultaneous counterparts. The simultaneous subgradient projections (SSP) method [32], [33] is a simultaneous variant of the CSP, and is given in Algorithm 2.

Algorithm 2 The method of simultaneous subgradient projections (SSP)

Initialization: $z^0 \in \mathbb{R}^n$ is arbitrary.

Iterative step:

i. Given z^k, calculate, for all $i \in I = \{1, 2, \ldots, p\}$, intermediate iterates $\zeta^{k+1,i}$ by

$$\zeta^{k+1,i} = \begin{cases} z^k - \alpha_k \dfrac{f_i(x^k)}{\parallel t^k \parallel^2} t^k, & \text{if } f_i(z^k) > 0, \\ z^k, & \text{if } f_i(z^k) \leq 0, \end{cases} \tag{16}$$

where $t^k \in \partial f_i(z^k)$ is a subgradient of f_i at the point z^k, and the relaxation parameters $\{\alpha_k\}_{k=0}^{\infty}$ are confined to an interval $\epsilon_1 \leq \alpha_k \leq 2 - \epsilon_2$, for all $k \geq 0$, with some, arbitrarily small, $\epsilon_1, \epsilon_2 > 0$.

ii. Calculate the next iterate z^{k+1} by

$$z^{k+1} = \sum_{i=1}^{p} w_i \zeta^{k+1,i} \tag{17}$$

where w_i are fixed, user-chosen, positive weights with $\sum_{i=1}^{p} w_i = 1$.

The convergence analysis for this algorithm is available only for consistent ($Q \neq \emptyset$) CFPs, see [32], [33].

C. CSP and SSP for Compressed Sensing

Based on Eqs. (4) and (5), we may formulate the following CS problem: Given a real-valued $m \times n$ matrix H, a measurement vector $y \in \mathbb{R}^m$ and a sparse (or compressible) vector $x \in \mathbb{R}^n$,

$$\text{find } x^* \in \mathbb{R}^n \text{ s. t. } y = Hx^*, \; \|x^*\|_1 \leq \epsilon \tag{18}$$

Eq. (18) can be translated into the language of CFP. To do this, we define m closed convex sets $Q_1, Q_2, \cdots, Q_m \subseteq \mathbb{R}^n$ expressed as

$$Q_i = \{x \in \mathbb{R}^n \mid y_i = \langle h_i, \, x \rangle\}, \tag{19}$$

where h_i indicates the i-th row of the matrix H. In this case, at a given iteration k, $t^k = \nabla \langle h_i, \, x \rangle = h_i$. An additional closed convex set $Q_{m+1} \subseteq \mathbb{R}^n$ is defined as

$$Q_{m+1} = \{x \in \mathbb{R}^n \mid \|x\|_1 - \epsilon \leq 0\} \tag{20}$$

For the latter convex set, we note that a subgradient at the origin can be chosen so that

$$t^k \in \partial \|x\|_1 \mid_{x=0} = 1_{1\times n} \tag{21}$$

where $1_{1\times n}$ is an n-dimensional vector of unit entries. It can be easily verified that the subgradient chosen in Eq. (21) satisfies the subgradient definition of Eq. (12). This subgradient may be generalized $\forall x \in \mathbb{R}^n$ by adopting the convention

$$\text{sign}(x_i) = \begin{cases} 1, & x_i \geq 0 \\ -1, & x_i < 0 \end{cases} \tag{22}$$

and writing, at some iteration k,

$$t^k = \text{sign}(x^k), \quad \|t^k\|_2^2 = n, \, \forall k. \tag{23}$$

The next stage is to implement the CSP algorithm in order to find a feasible solution $x^* \in Q := \cap_{i=1}^{m+1} Q_i$ for the CFP formulation of the CS problem (18). Following the recipe of Algorithm 1, using the relationships (21), (22) and (23), such an algorithm is formulated as follows:

An application of Algorithm 2 for the CS problem (18), termed SSP-CS, can be done in a similar manner to Algorithm 3, and is omitted here for the sake of conciseness.

Note that the inactive step of Algorithms 1 and 2, i. e., when the current iterate remains unchanged, is unnecessary if the constraints are linear equations (cf. [30, Chapter 6]). This fact helps increasing the computational efficiency. In the same context, it is worth noting that without using the 1-norm constraint implemented in Eq. (25), the recovery of x from the linear set of equations $y = Hx$ is possible, under certain regularity conditions, using the well-known algorithm of Kaczmarz [30, Chapter 6]. In this case, the obtained solution x^* would minimize $\|x\|_2$. However, adding the 1-norm constraint is key to a successful recovery when x is sparse (or compressible).

Algorithm 3 Cyclic subgradient projections for compressed sensing (CSP-CS)

Initialization: $\hat{x}^0 \in \mathbb{R}^n$ is arbitrary.

Constraint-Index Control: Set $i(k) = k(\mod p) + 1$ for all $k \geq 0$, where $p = m + 1$.

Iterative step: For $1 \leq i \leq m$, calculate $\hat{x}^{k+1}$ by

$$\hat{x}^{k+1} = \hat{x}^k - \alpha^k \frac{\langle h_i,\, \hat{x}^k \rangle - y_i}{\parallel\ h_i \parallel_2^2} h_i \tag{24}$$

with the relaxation parameters $\{\alpha^k\}_{k=0}^{\infty}$ confined to the open interval $(0,\, 2)$.

For $i = p = m + 1$, calculate $\hat{x}^{k+1}$ by

$$\hat{x}^{k+1} = \hat{x}^k - \lambda^k \left[\frac{\|\hat{x}^k\|_1 - \epsilon}{n} \text{sign}(\hat{x}^k) \right] \mathcal{H}[\|\hat{x}^k\|_1 - \epsilon] \tag{25}$$

where $\mathcal{H}[x]$ denotes the standard step function (i.e., $\mathcal{H}[x] = 1$ for $x > 0$ and $\mathcal{H}[x] \leq 0$ otherwise), and $\{\lambda^k\}_{k=0}^{\infty}$ are confined to the open interval $(0,\, 2)$.

The convergence of Algorithms 3 and its SSP variant to some feasible solution x^* is guaranteed, provided that the CFP is consistent (i. e., the solution set is non-empty). However, for real-world, large-scale problems, one cannot determine *a-priori* whether a given value of ϵ renders the CFP consistent; this is essentially a matter of tuning. The behavior of the subgradient projections algorithms for the inconsistent case was investigated in Ref. [26], where it was shown that simultaneous subgradient projections methods usually yield better convergence in the inconsistent case. However, simultaneous projection methods tend to converge slower than sequential methods; it is thus a good idea to combine these two methods in order to fight instabilities and obtain fast convergence. This issue is discussed in the next section.

IV. Practical Implementation

In this section we discuss several issues concerning the practical implementation of the CSP-CS and SSP-CS algorithms.

A. *Stopping Criteria*

The CSP-CS and SSP-CS routines are terminated when either some predetermined maximal number of iterations is exceeded, or when there is no significant difference between two consecutive estimates, viz. $\parallel \hat{x}^{k+1} - \hat{x}^k \parallel_2 \leq \gamma$ for some small $\gamma > 0$.

B. *Refinements*

As mentioned in the previous section, in some cases the set of constraints may not be consistent due to an improper setting of ϵ, which in turn implies that the convergence of the CSP method is not guaranteed. Typically, in such scenarios the estimated parameters persistently fluctuate around some nominal value. The reconstructed signal itself can be highly accurate; however, its entries do not attain a fixed value. This problem can be alleviated by incorporating an additional refinement stage. In this work, we propose two such refinement approaches, each of which relies on a different technique for producing refined estimates.

1) Gauss-CSP: After a given number of CSP-CS iterations, the support of the unknown signal can be readily approximated based on the magnitudes of the estimated elements. As soon as the support is given, the signal can be accurately reconstructed following a simple least squares (LS) procedure. This approach is inspired by the strategy adopted in [8] for correcting the bias inherent to the Dantzig selector (the corrected scheme was referred to in [8] as the Gauss-Dantizg selector). This technique, applied to the CSP-CS algorithm, will be referred to as the *Gauss-CSP*. This variant of CSP-CS is summerized in Algorithm 4.

Algorithm 4 Gauss-CSP

CSP-CS Stage: Perform several CSP-CS iterations based on Algorithm 3 until a stopping criteria is reached.

LS Stage: Set $N < m$ as the maximal support size of the obtained estimate $\hat{x}^k$. Define a set of indices $G := \{j_i\}_{i=1}^N$ of the most significant entries in $\hat{x}^k$ in terms of magnitude. Solve

$$\beta = \left(\hat{H}^T \hat{H}\right)^{-1} \hat{H}^T y \tag{26}$$

where $\hat{H} \in \mathbb{R}^{m \times N}$ is composed of the columns of H corresponding to the indices in G. Set $\hat{x}_l^k = 0$, $\forall l \notin G$ and $\hat{x}_{j_i}^k = \beta_i$, $i = 1, \dots, N$.

A slightly different implementation of the Gauss-CSP, referred to as *alternating Guass-CSP*, can dramatically improve the convergence properties of the CSP method. This consists of incorporating the LS stage directly into the CSP algorithm in an alternating manner. This variant of the Gauss-CSP typically converges much faster than any other alternative. Following this approach, the LS routine is executed whenever the cyclic index i in Algorithm 3 reaches an arbitrary predetermined value.

2) CSP-SSP: Normally, running a few SSP iterations subsequent to the termination of the CSP routine improves the accuracy of recovery. This stems from the improved behavior of simultaneous subgradient projections method in the inconsistent case, discussed in the previous section.

C. Block Processing

In some programming environments, it might be more computationally efficient to process a group of l observations $y_i = \langle h_i, x \rangle$, $i = j, \dots, j + l$ at each iteration rather than a single one. Our own implementation of the CSP-CS exploits this simple idea for alleviating the workload on the MATLAB®

interpreter, which may become prohibitively slow when loops are involved. This approach does not require any considerable modification of the original algorithm. In practice, this is accomplished by running the cyclic index from 1 to $p/l+1$ while repeating the CSP update stage (24) l times at each iteration (one update for each observation in the block).

V. Illustrative Examples

In this section, the new convex feasibility programming algorithms are assessed based on an extensive comparison with some of the commonly-used methods for CS. The algorithms considered herein consist of the homotopy method LARS [7], the two greedy algorithms, OMP [18] and BP [9], the recently-introduced gradient projection (GP)-based method of [20], and the Bayesian CS (BCS) of [15] (the links for the MATLAB® implementations of the various methods used here are provided in the Appendix). In order to highlight the weaknesses and virtues of the various methods, we examine here both synthetic and realistic scenarios, associated with the two signal types frequently encountered in CS applications: Sparse and compressible (nearly sparse in the sense of (8)). An indicator for the difficulty of recovery is assigned for each problem based on its unique settings (dimension and sparseness degree). This measure, termed here the recovery index, is derived from the relation $m \geq c(s \log n)$ [1] (for some $c > 0$) as

$$(s/m) \log n \tag{27}$$

This index essentially refers to the probability of recovery assuming the sensing matrix obeys the RIP up to a certain sparseness degree. As it was already pointed out in [1], as this measure increases an exact recovery becomes less probable.

Throughout this section, the signal reconstruction error is computed as

$$e := \sqrt{\frac{\| x - \hat{x} \|_2^2}{d(x)}} \tag{28}$$

where the normalizing term $d(x)$ is set according to the signal type,

$$d(x) := \begin{cases} \sum_{i=1}^{n} \min(x_i^2, \sigma^2), & \text{for sparse } x \\ \| x \|_2^2, & \text{for compressible } x \end{cases} \tag{29}$$

with σ being the standard deviation of the observation noise. The above formulation consolidates both the *ideal* and the *normalized* recovery measures that are used for assessing the reconstruction accuracy in [8] and [15], respectively. The mean reconstruction error is computed in a similar fashion over N runs via $\sqrt{N^{-1} \sum_i e^2(i)}$, where $e(i)$ denotes the error in the ith run.

A. Least Squares Augmentation

The approach underlying the Gauss-Dantzig [8] and the Gauss-CSP in Algorithm 4, which essentially consists of incorporating an additional LS stage for refining the obtained estimates, can be easily applied for any other CS method. This poses a question as to what should be a fair comparison when considering the Gauss-CSP variant. Intuitively, one could think that the relative improvement of the LS stage as described in Algorithm 4 would be identical irrespectively of the CS method, and therefore such variants should not be compared at all. This statement, however, turns out to be incorrect, as shown by our experiments. In order to maintain our observations as equitable as possible in these circumstances, our comparison involves, in addition, an LS-augmented variant of each of the aforementioned CS techniques.

B. *Synthetic Example*

In the first scenario, the various methods are applied for the recovery of a sparse signal from noisy observations. The sensing matrix $H \in \mathbb{R}^{m \times n}$ used here consists of normalized random entries sampled from a zero-mean Gaussian distribution with a standard deviation of $1/\sqrt{m}$. This type of random construction has been shown to obey the RIP up to a reasonable sparseness level (see [2]). We examine the recovery performance of the CS algorithms in various settings consisting of different problem dimensions, ranging from 512x1024 (small) through 1024x2048 and 2048x4096 (medium) to 3072x6144 (large), and different sparseness levels. The original signal x is composed of only few nonzero random elements, which are uniformly sampled over $[-1, 1]$, and of which the indices are randomly picked between 1 and n. In all runs, the measurement noise standard deviation is set as 0.01. The various algorithms' tuning parameters are taken as those that seemed to minimize the recovery error based on tuning runs. The CSP-CS relaxation parameters are set as

$$\alpha^k = 1.8, \quad \epsilon = 10^{-4}, \quad \frac{\lambda^k}{n} = \begin{cases} 70^{-2}, & \text{If } k \leq 2000 \\ 100^{-2}(1 + k/10^4)^{-1}, & \text{Otherwise} \end{cases} \tag{30}$$

which seemed to be the best values with respect to accuracy and convergence time. The CSP routine is terminated when either the number of iterations exceeds 5000 or when the normed difference between two consecutive estimates, $\| \hat{x}_{k+1} - \hat{x}_k \|_2$, drops below some γ where $\gamma = 0.01$ for the low dimensional problem and $\gamma = 0.5$ otherwise.

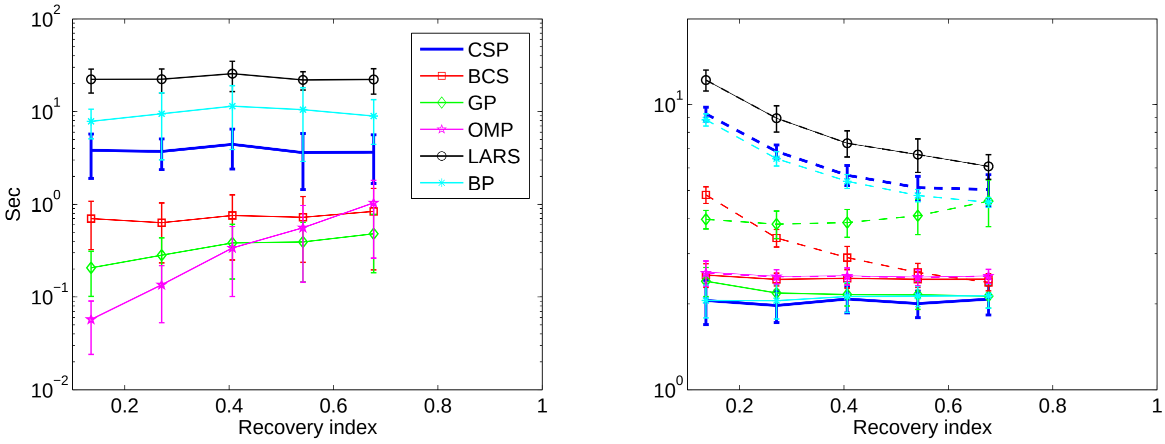
Both the OMP and the LS refinement stage assume a maximal support size of the reconstructed signal. In all our experiments we have chosen this parameter as $1.5s$ where s is the actual support size (sparseness level) of x.

The averaged performance over 50 Monte Carlo runs of the various methods in the small-scale scenario is depicted in Fig. 1 for different recovery indices ranging from 0.1 to 0.7. This figure shows both the mean ideal recovery error and the corresponding mean convergence time along with their standard deviations, which are illustrated using error bars. Observing the right panel in this figure reveals a performance hierarchy that places the CSP as the 2nd worst right after the LARS and just a bit before the BP. The remaining methods attain higher accuracy in this case.

Nevertheless, the story completely changes when examining the LS-augmented methods. Here the Gauss-CSP attains the *best accuracy* over the entire range of sparseness degrees. The advantage of using the LS stage is likewise prominent in all other methods, as it significantly reduces the recovery error. By observing the left panel of Fig. 1, it can be easily recognized that the CSP is the 3rd slowest method in this case. The other computationally excessive methods here are the LARS and the BP. In this small-scale example, the convergence time of all methods excluding those of the greedy OMP and the GP roughly stays unchanged over the entire range of sparseness levels. The extreme slant of the OMP line is due to the nature of this algorithm, which tends to become computationally intensive as the assumed maximal support size increases.

Note that the convergence times of the LS-augmented methods are omitted here and in the sequel, as they are nearly identical to those of the original unaugmented methods (the LS stage is implemented using the `MATLAB`® pseudo-inverse command `pinv`, which is extremely fast even in high dimensions).

The above insights are further emphasized in Table I, which repeats the values from Fig. 1 for two nominal recovery indices. The bold values in this table correspond to the averaged recovery errors of the LS-augmented methods. Thus, it can be clearly seen that the Gauss-CSP attains the best accuracy yielding a mean recovery error of around 2.05.



(a) Convergence Time

(b) Ideal normalized error

Fig. 1: The average performance (over 50 Monte Carlo runs) of the CS methods (including the computationally excessive ones, the LARS and the BP) for problem dimension 512x1024. The performance of the LS-augmented variants is depicted via solid lines.

The performance of the algorithms in the medium and large scale scenarios is illustrated in Fig. 2. Here we have excluded the computationally intensive methods, the LARS and the BP, as their

Method	$(s/m)\log n = 0.1$		$(s/m)\log n = 0.6$		$(s/m)\log n = 0.1$	$(s/m)\log n = 0.6$
LARS	12.19	**12.19**	6.07	**6.07**	22.28 (sec)	22.18 (sec)
BP	8.84	**2.05**	4.54	**2.13**	7.84 (sec)	8.93 (sec)
OMP	2.56	**2.58**	2.50	**2.50**	0.05 (sec)	1.03 (sec)
BCS	4.82	**2.52**	2.37	**2.44**	0.70 (sec)	0.83 (sec)
GP	3.95	**2.40**	4.58	**2.12**	0.20 (sec)	0.48 (sec)
CSP	9.28	**2.05**	5.04	**2.07**	3.81 (sec)	3.64 (sec)

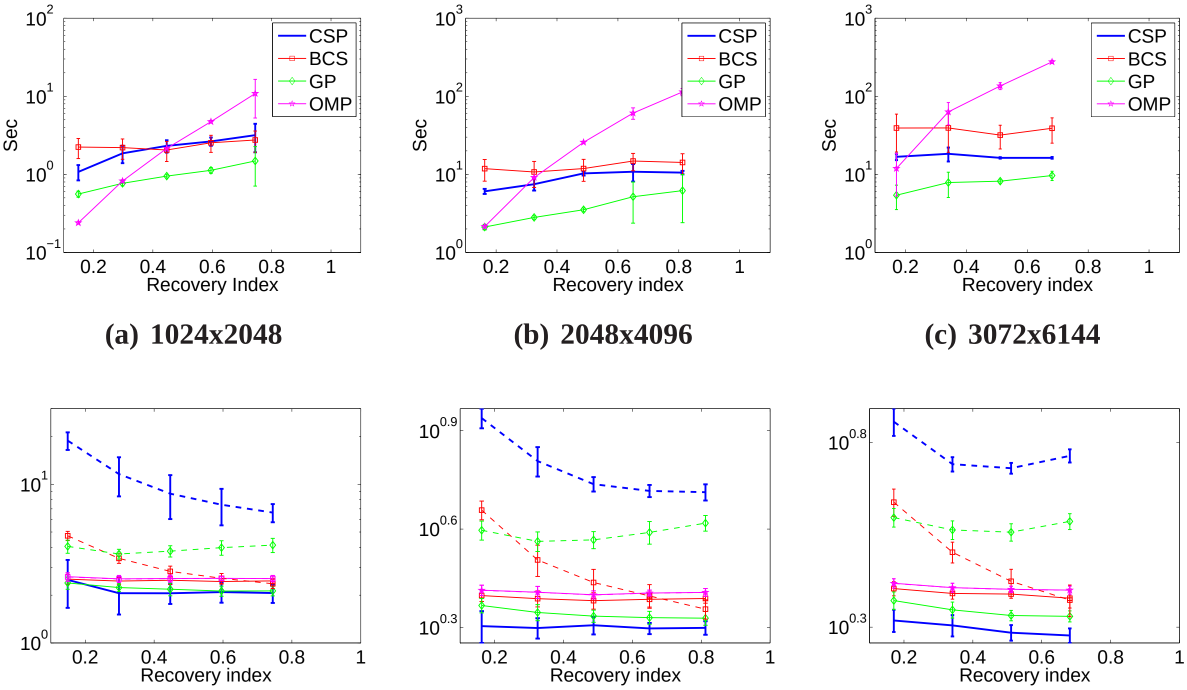
TABLE I: Recovery of sparse signals. The ideal recovery error (left columns) and convergence time (right columns) of the various methods for the problem dimension 512x1024. The bold values correspond to the accuracy of the two-staged LS-augmented variants. Averaged over 50 Monte Carlo runs.

convergence time became prohibitively long. The performance of the remaining methods for the various problem dimensions is illustrated via two panels. Thus, the upper panel shows the convergence times for different sparseness levels, whereas the bottom panel shows the corresponding ideal recovery errors averaged over 50 Monte Carlo runs. As before, the standard deviations from the mean values are depicted using error bars.

By observing the upper panel in this figure, it can be easily recognized that the CSP method is comparable in speed and even faster than the BCS over the entire range of sparseness levels in the medium scale scenarios. A similar conclusion applies when comparing the CSP with the OMP from a certain sparseness degree corresponding to a recovery index of around 0.4. The CSP turns out to be *significantly faster* than both the BCS and the OMP over almost the entire range of sparseness levels in the large scale scenario.

The corresponding recovery errors of the various methods are presented in the bottom panel in this figure. Thus, it can be recognized that the unaugmented CSP yields a slightly worse accuracy than the other methods (notice the logarithmic ordinate). The 2nd worse method in terms of accuracy in this case is also the fastest of them all, the GP. Nevertheless, when looking at the augmented methods it turns out that, as we have already witnessed in the small-scale scenario, the Gauss-CSP is the best method in terms of accuracy. The 2nd best in this case is the LS-augmented GP.

The above insights are supported by Table II, which provides the timing and recovery error values in the large scale scenario for two nominal recovery indices. Thus, it can be easily seen that the Gauss-CSP outperforms the other methods in terms of accuracy and is also the 2nd fastest method (after the GP). Another interesting and important detail that stems from both Table II and Fig. 2 is related to the fact that as opposed to the GP, of which the running time is highly sensitive to the sparseness degree, the CSP computation time remains almost unchanged with respect to this factor. This observation could, in fact, be expected, as the CSP mechanism does not really distinguish between elements in the support and those which are not. This renders it highly robust and efficient when applied either to compressible problems or in such scenarios where the recovery index is relatively large.



(a) 1024x2048 **(b) 2048x4096** **(c) 3072x6144**

(d) 1024x2048 **(e) 2048x4096** **(f) 3072x6144**

Fig. 2: The average performance of the CS algorithms for various problem dimensions and sparseness degrees. Showing the convergence time (upper panel) and the ideal recovery error (bottom panel). The computationally excessive methods (LARS and BP) are not shown here.

A comparison of the various CSP implementations that were discussed in Section IV-B is provided in Fig. 3. This figure demonstrates the convergence properties of the methods based on a single run for a problem dimension of 2048x4096. Thus, it can be easily recognized that the plain CSP attains the worst recovery error, as it begins to fluctuate around some nominal value. As was pointed out previously, this behavior indicates that the set of constraints in this case is inconsistent owing to an improper setting of the parameter ϵ. This problem is alleviated in both of the variants, the CSP-SSP and the Gauss-CSP. Although both these methods outperform the plain CSP, it seems that the best attainable error is achieved by using the Gauss-CSP, whereas the fastest convergence is obtained by using the alternating CSP-LS scheme.

Method	$(s/m)\log n = 0.1$		$(s/m)\log n = 0.6$		$(s/m)\log n = 0.1$	$(s/m)\log n = 0.6$
OMP	2.62	**2.62**	2.52	**2.52**	11.89 (sec)	135.52 (sec)
BCS	4.35	**2.53**	2.65	**2.45**	39.18 (sec)	31.84 (sec)
GP	3.95	**2.35**	3.60	**2.14**	5.41 (sec)	8.18 (sec)
CSP	7.18	**2.08**	5.37	**1.92**	16.79 (sec)	16.26 (sec)

TABLE II: Recovery of sparse signals. The ideal recovery error (left columns) and convergence time (right columns) of the various methods for the problem dimension 3072x6144. The bold values correspond to the accuracy of the two-staged LS-augmented variants. Averaged over 50 Monte Carlo runs.

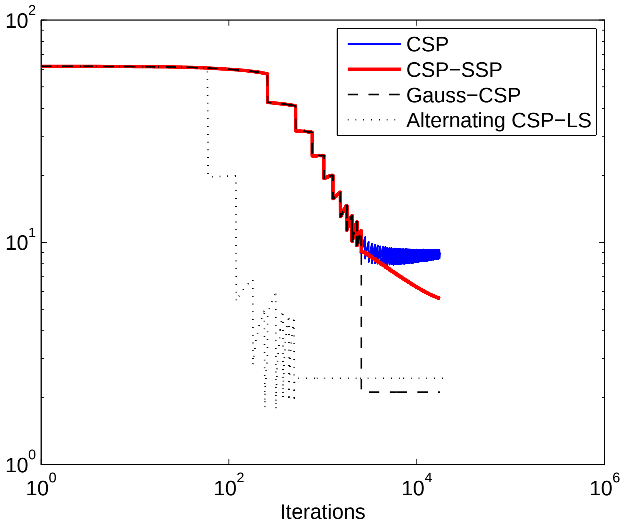


Fig. 3: The ideal recovery error of the CSP variants. Single run, problem dimension 2048x4096.

C. *Realistic (Compressible) Examples*

Compressible signals are of greater practical importance when it comes to real-world applications. As was previously pointed out, such signals are nearly sparse in the sense that they consist of many relatively small elements that are not identically zero. Following this, we consider here two realistic experimental studies involving compressible signals. The first example, which is conducted in the spirit of the previous synthetic one, consists of constructing the discrete Fourier transform (DFT) of an undersampled time series. The second experiment, which follows right after, involves the reconstruction of the famous Shepp-Logan phantom head image that is commonly used for assessing the performance of recovery schemes in tomography.

1) Constructing a DFT from Undersampled Data: In this example we consider a discrete signal in the time domain, which takes the form

$$y_k = \sum_{i=1}^{n_f} \sin(\omega_i t_k) \tag{31}$$

where the frequencies ω_i, $i = 1, \ldots, n_f$ are uniformly sampled over $[1, 10\pi]$. Let $x \in \mathbb{R}^n$ be the DFT of y_k over the discrete times $k = 1, \ldots, n$, that is

$$x_k = 1/\sqrt{n} \sum_{j=1}^{n} y_j \exp\left(-2\pi(j-1)(k-1)i/n\right) \tag{32}$$

which can be written compactly as $x = Fy$ with F being the unitary DFT matrix. Now, suppose that we wish to reconstruct x from a randomly sampled set of points $\{y_j\}_{j=1}^{m}$ for which $m << n$. In other words, we attempt to compose the DFT x from an undersampled time series. This is a classical CS problem, for which the sensing matrix H is given as a partial inverse DFT matrix consisting of only m randomly-picked rows. Thus,

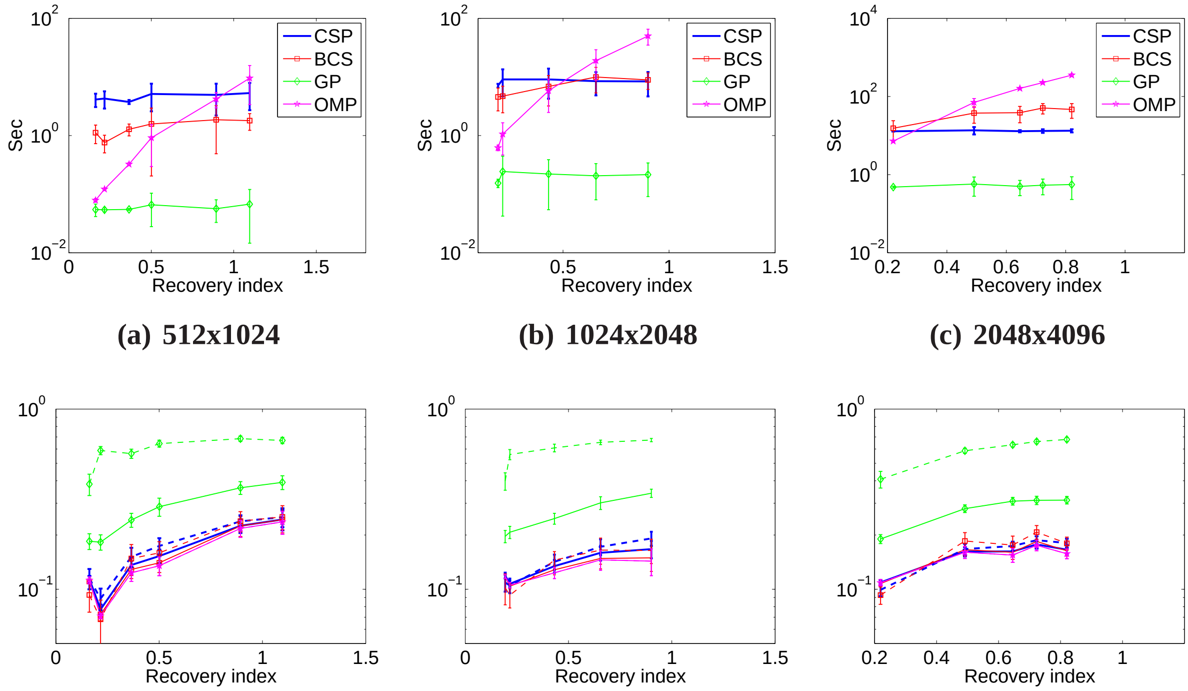
$$y = Hx \tag{33}$$

where H is composed of m randomly picked rows from F^*, corresponding to the time points in y. The vector x itself is expected to be compressible, essentially comprised of decaying coefficients in the vicinity of those which are associated with the underlying frequencies ω_i, $i = 1, \ldots, n_f$.

Similarly to the synthetic case, we apply the various CS methods (excluding the computationally excessive ones) for different problem dimensions and sparseness levels. As distinct from the previous example, here the recovery index (27) is computed based on the *effective* sparseness measure (8) with $\varepsilon = 0.05 \max_i |x_i|$. The CS algorithms' tuning parameters are chosen to maximize accuracy based on tuning runs. The CSP relaxation variables and termination conditions remain unchanged.

The averaged performance of the various methods over 50 Monte Carlo runs (in which a new set of frequencies $\{w_i\}_{i=1}^{n_f}$ is sampled at the beginning of each run) is depicted in Fig. 4. The upper panel in this figure shows the CS methods' mean convergence times and their associated standard deviations (error bars) for different problem dimensions (ranging from 512x1024 through 1024x2048 to 2048x4096) and effective sparseness levels (corresponding to recovery indices of between 0.1 to 0.8). The corresponding recovery errors of the various methods are provided in the bottom panel of this figure.

By observing both these panels, it can be clearly recognized that the CSP method maintains the *best tradeoff* between accuracy and computational load as the problem becomes more complex (as indicated by both its dimensionality and sparseness degree). In virtue of its underlying mechanism, the CSP exhibits robustness with respect to both these factors as it attains recovery errors that are comparable to those of the BCS and the OMP at a nearly fixed computational cost. This renders it the fastest reliable method as the problem dimension increases. Although the GP is the fastest scheme in this case, its accuracy is extremely low with recovery errors of nearly the magnitude of the signal itself. The LS-augmented GP does posses a clear advantage over the unaugmented one; however, its performance is still unsatisfactory (with recovery errors of almost twice than those of the other methods).



(a) 512x1024 (b) 1024x2048 (c) 2048x4096

(d) 512x1024 (e) 1024x2048 (f) 2048x4096

Fig. 4: Recovering a DFT from undersampled data. Showing the average performance of the CS algorithms for various problem dimensions and effective sparseness degrees. The convergence time and the normalized error are depicted in the upper and lower panels, respectively.

The above insights are further illustrated in both Tables III and IV in which the timing and recovery error values are repeated for the small and large scale scenarios, respectively. Thus, it can be seen from the first table (small scale) that the CSP attains a recovery error that is similar to those of the BCS and the OMP. Its convergence time, however, is longer compared to the other methods. The prominent advantage of the CSP is manifested in the 2nd table (large scale) in which it attains the best performance in terms of accuracy and convergence time for a relatively large recovery index of 0.8. Its robustness to the sparseness level is illustrated in both these tables by the nearly fixed convergence time for the two extremal values of the recovery index.

Figure 5 depicts the performance of the CSP in recovering typical sparse and compressible signals of nearly the same recovery index. This figure suggests that, in practice, though the recovery indices are nearly the same, it might be more difficult to reconstruct a compressible representation rather than

Method	$(\hat{s}/m)\log n = 0.2$		$(\hat{s}/m)\log n = 0.8$		$(\hat{s}/m)\log n = 0.2$	$(\hat{s}/m)\log n = 0.8$
OMP	0.11	**0.11**	0.21	**0.21**	0.07 (sec)	4.19 (sec)
BCS	0.09	**0.11**	0.23	**0.22**	1.11 (sec)	1.86 (sec)
GP	0.38	**0.18**	0.68	**0.36**	0.05 (sec)	0.05 (sec)
CSP	0.11	**0.11**	0.23	**0.22**	4.11 (sec)	4.93 (sec)

TABLE III: Recovering a DFT from undersampled data. The normalized recovery error (left columns) and convergence time (right columns) of the various methods for the problem dimension 512x1024. The bold values correspond to the accuracy of the two-staged LS-augmented variants. Averaged over 50 Monte Carlo runs.

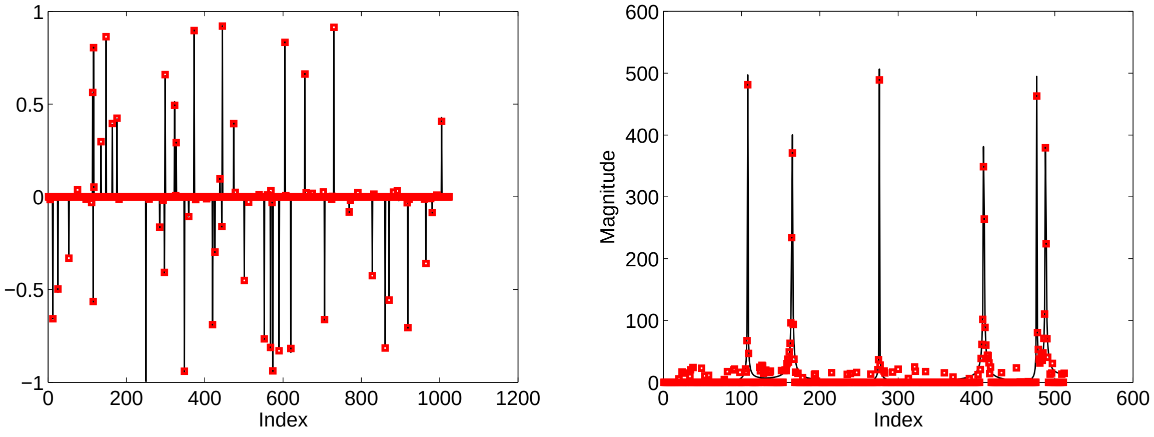
Method	$(\hat{s}/m)\log n = 0.2$		$(\hat{s}/m)\log n = 0.8$		$(\hat{s}/m)\log n = 0.2$	$(\hat{s}/m)\log n = 0.8$
OMP	0.10	**0.10**	0.17	**0.17**	7.20 (sec)	226.80 (sec)
BCS	0.09	**0.10**	0.20	**0.18**	15.40 (sec)	50.62 (sec)
GP	0.40	**0.19**	0.66	**0.31**	0.47 (sec)	0.53 (sec)
CSP	0.09	**0.10**	0.18	**0.17**	12.87 (sec)	13.11 (sec)

TABLE IV: Recovering a DFT from undersampled data. The normalized recovery error (left columns) and convergence time (right columns) of the various methods for the problem dimension 2048x4096. The bold values correspond to the accuracy of the two-staged LS-augmented variants. Averaged over 50 Monte Carlo runs.

a sparse one. This follows from the fact that the compressible estimate on the right sub-figure is less accurate than its companion on the left.

2) Image Recovery from Undersampled Radial Fourier Coefficients: In the last part of this section, we demonstrate the performance of some of the methods in recovering an image using undersampled 2D Fourier coefficients that are computed along radial lines. The example considered here follows along the lines of [8], where the Shepp-Logan phantom head image is used. In our experiment, however, we use a low-dimensional 128x128 version of this image, which yields a signal of dimension $128^2 = 16384$. We examine the CS methods for two scenarios in which the 2D Fourier coefficients are sampled along either 32 or 64 radial lines (corresponding to nearly 25% or 50% of the available data).

Three methods are applied for recovering the phantom head image: BCS, GP and CSP (unaugmented versions). However, the performance of only two of them, the GP and the CSP are shown in Fig. 6 as the BCS exhibited poor performance, essentially yielding recovery errors of nearly the signal magnitude (specifically, around 0.9). Figure 6 clearly shows the superiority of the CSP over the GP



(a) Sparse signal

(b) DFT (compressible) signal

Fig. 5: Illustration of the recovery performance of the CSP method for sparse and compressible signals.

in both scenarios.

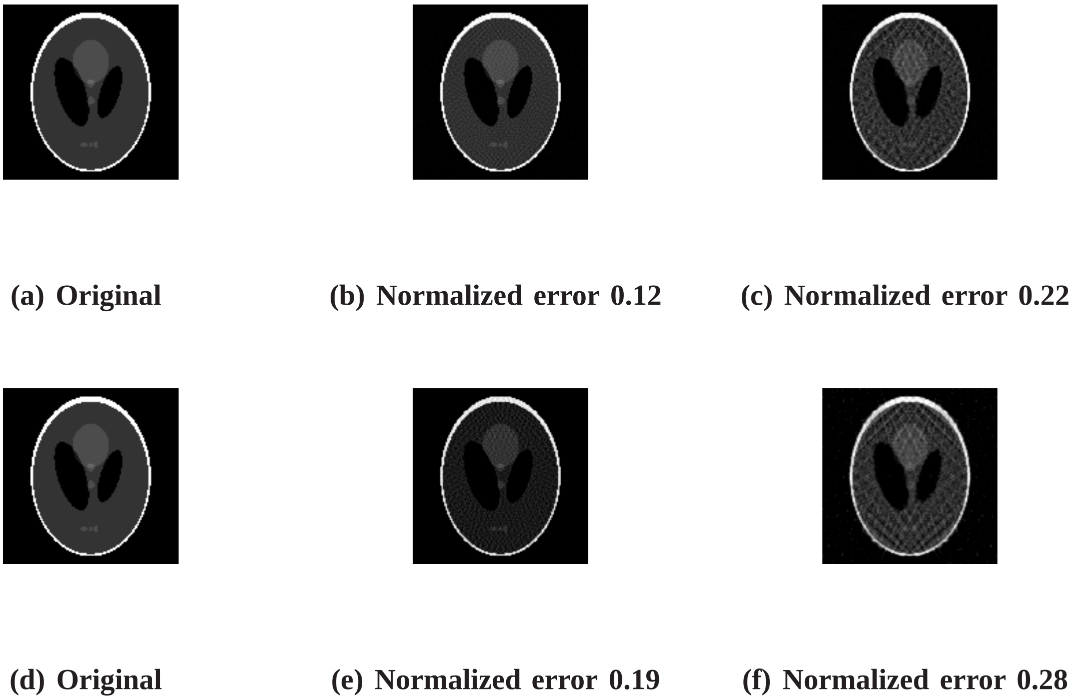
VI. Conclusions

A novel compressed sensing (CS) scheme was introduced for solving large-scale compressible problems. The new method utilizes the instantaneous subgradient for projecting the previous iterate on some intermediate point, thereby approaching the underlying convex feasibility set resulting from a group of possibly nonlinear constraints. The cyclic subgradient projection (CSP) mechanism, which is the heart of the new approach, facilitates the efficient solution of large-scale CS problems owing to its implementation, which involves vector products only. An extensive numerical comparison of the CSP-CS algorithm and its variants with some of the state-of-the-art CS schemes clearly demonstrated its superiority in high-dimensional compressible settings.

In particular, we may draw the following conclusions. First, the newly-proposed method maintains a nearly fixed computational cost, irrespectively to the problem dimension and sparseness level; moreover, the new method easily copes with realistic scenarios involving high dimensional compressible signals.

Second, a de-biased CSP, which utilizes an additional least-squares stage, improves the performance for sparse signals, showing superiority compared to other de-biased methods.

Finally, the extensive numerical comparison of all the methods clearly shows that the CSP maintains the best tradeoff between computational efficiency and accuracy as the problem dimension and sparseness level increase. As such, we conclude that the CSP is the most efficient method for large-scale compressible problems.



(a) Original **(b) Normalized error 0.12** **(c) Normalized error 0.22**

(d) Original **(e) Normalized error 0.19** **(f) Normalized error 0.28**

Fig. 6: The original and reconstructed 128x128 Shepp-Logan phantom head. Showing the CSP (upper panel) and the GP (lower panel) methods. The middle and right rows depict the recovery error based on 64 and 32 projections, respectively (equivalent to 50% and 25% undersampled data).

ACKNOWLEDGMENTS

This research was partially supported by the Asher Space Research Institute. The Authors wish to acknowledge Yair Censor, whose seminal work on optimization inspired the research described herein.

APPENDIX

The `MATLAB`® files implementing our new algorithms are available at `http://www.technion.ac.il/ pgurfil/csp-cs/`. All the other source files for the various CS algorithms implemented herein can be found at the following locations:

- Least Angle Regression (LARS) :
  `http://www.mathworks.com/matlabcentral/fileexchange/23186-lars-algorithm`
- Basis Pursuit (BP):
  `http://www.acm.caltech.edu/l1magic/`
- Bayesian CS (BCS):
  `http://people.ee.duke.edu/ lihan/cs/`

- Gradient Projection for Sparse Reconstruction (GPSR):
`http://www.lx.it.pt/ mtf/GPSR/`

REFERENCES

[1] E. J. Candes, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information”, *IEEE Transactions on Information Theory*, vol. 52, pp. 489–509, 2006.

[2] E. J. Candes, “Compressive sampling”, Madrid, Spain, 2006, European Mathematical Society, Proceedings of the International Congress of Mathematicians.

[3] M. Lustig, D. Donoho, and J. M. Pauly, “Sparse MRI: The application of compressed sensing for rapid MR imaging”, *Magnetic Resonance in Medicine*, vol. 58, pp. 1182–1195, 2007.

[4] U. Gamper, P. Boesiger, and S. Kozerke, “Compressed sensing in dynamic MRI”, *Magnetic Resonance in Medicine*, vol. 59, pp. 365–373, 2008.

[5] R. Chartrand, “Exact reconstruction of sparse signals via nonconvex minimization”, *IEEE Signal Processing Letters*, vol. 14, pp. 707–710, 2007.

[6] R. Tibshirani, “Regression shrinkage and selection via the LASSO”, *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[7] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, “Least angle regression”, *Annals of Statistics*, vol. 32, no. 2, pp. 407 – 499, 2004.

[8] E. Candes and T. Tao, “The Dantzig selector: statistical estimation when p is much larger than n”, *Annals of Statistics*, vol. 35, pp. 2313–2351, 2007.

[9] S. S. Chen, D. L. Donoho, and M. A. Saunders, “Atomic decomposition by basis pursuit”, *SIAM Journal of Scientific Computing*, vol. 20, no. 1, pp. 33 – 61, 1998.

[10] M. E. Tipping, “Sparse Bayesian learning and the relevance vector machine”, *Journal of Machine Learning Research*, vol. 1, pp. 211 – 244, 2001.

[11] R. E. McCulloch and E. I. George, “Approaches for Bayesian variable selection”, *Statistica Sinica*, vol. 7, pp. 339 – 374, 1997.

[12] J. Geweke, *Bayesian Statistics 5*, chapter Variable selection and model comparison in regression, Oxford University Press, 1996.

[13] B. A. Olshausen and K. Millman, “Learning sparse codes with a mixture-of-Gaussians prior”, *Advances in Neural Information Processing Systems (NIPS)*, pp. 841 – 847, 2000.

[14] S. J. Godsil and P. j. Wolfe, “Bayesian modelling of time-frequency coefficients for audio signal enhancement”, *Advances in Neural Information Processing Systems (NIPS)*, 2003.

[15] S. Ji, Y. Xue, and L. Carin, “Bayesian compressive sensing”, *IEEE Transactions on Signal Processing*, vol. 56, pp. 2346 – 2356, June 2008.

[16] A. Carmi, P. Gurfil, and D. Kanevsky, “Methods for sparse signal recovery using kalman filtering with embedded pseudo-measurement norms and quasi-norms”, *IEEE Transactions on Signal Processing. Accepted*.

[17] S. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries”, *IEEE Transactions on Signal Processing*, vol. 4, pp. 3397 – 3415, 1993.

[18] Y. C. Pati, R. Rezifar, and P. S. Krishnaprasad, “Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition”, 27th Asilomar Conf. on Signals, Systems and Comput., 1993.

[19] S. Chen, S. A. Billings, and W. Luo, “Orthogonal least squares methods and their application to non-linear system identification”, *International Journal of Control*, vol. 50, pp. 1873 – 1896, 1989.

[20] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, “Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems”, *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, pp. 586 – 597, December 2007.

[21] P. L. Combettes, “The foundations of set-theoretic estimation”, *Proceedings of the IEEE*, vol. 81, pp. 182–208, 1993.

[22] P. L. Combettes, “The convex feasibility problem in image recovery”, *Advances in Imaging and Electron Physics*, vol. 95, pp. 155–270, 1996.

[23] G. T. Herman, *Image reconstruction from projections: The fundamentals of computerized tomography*, Academic Press, New York, NY, USA, 1980.

[24] Y. Censor, M. D. Altschuler, and W. D. Powlis, “On the use of Cimmino’s simultaneous projections method for computing a solution of the inverse problem in radiation therapy treatment planning”, *Inverse Problems*, vol. 4, pp. 607–623, 1988.

[25] L. D. Marks, W. Sinkler, and E. Landree, “A feasible set approach to the crystallographic phase problem”, *Acta Crystallographica*, vol. A55, pp. 601–612, 1999.

[26] D. Butnariu, Y. Censor, P. Gurfil, and E. Hadar, “On the behavior of subgradient projections methods for convex feasibility problems in Euclidean spaces”, *SIAM Journal on Optimization*, vol. 19, no. 2, pp. 786–807, 2008.

[27] I. Yamada, “Hybrid steepest descent method for variational inequality problem over the fixed point set of certain quasi-nonexpansive mappings”, *Numerical Functional Analysis and Optimization*, vol. 25, pp. 619–655, 2004.

[28] G. Crombez, “Non-monotoneous parallel iteration for solving convex feasibility problems”, *Kybernetika*, vol. 39, pp. 547–560, 2003.

[29] G. Crombez, “A sequential iteration algorithm with non-monotoneous behaviour in the method of projections onto convex sets”, *Czechoslovak Mathematical Journal*, vol. 56, pp. 491–506, 2006.

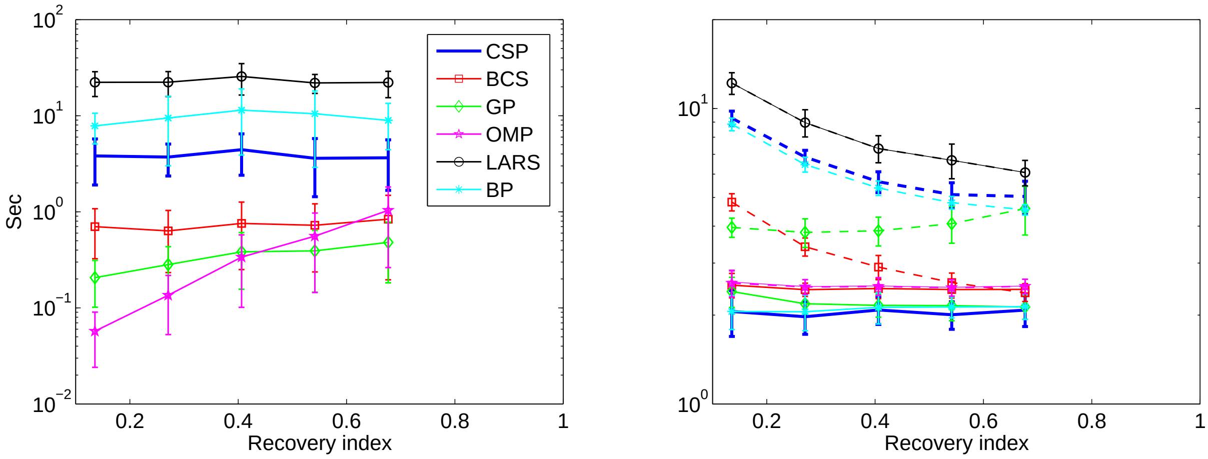
[30] Y. Censor and S. A. Zenios, *Parallel optimization: Theory, algorithms, and applications*, Oxford University Press, New York, NY, USA, 1997.

[31] Y. Censor and A. Lent, “Cyclic subgradient projections”, *Mathematical Programming*, vol. 24, pp. 233–235, 1982.

[32] A. N. Iusem and L. Moledo, “A finitely convergent method of simultaneous subgradient projections for the convex feasibility problem”, *Computational and Applied Mathematics*, vol. 5, pp. 169–184, 1986.

[33] L. T. Dos Santos, “A parallel subgradient method for the convex feasibility problem”, *Journal of Computational and Applied Mathematics*, vol. 18, pp. 307–320, 1987.

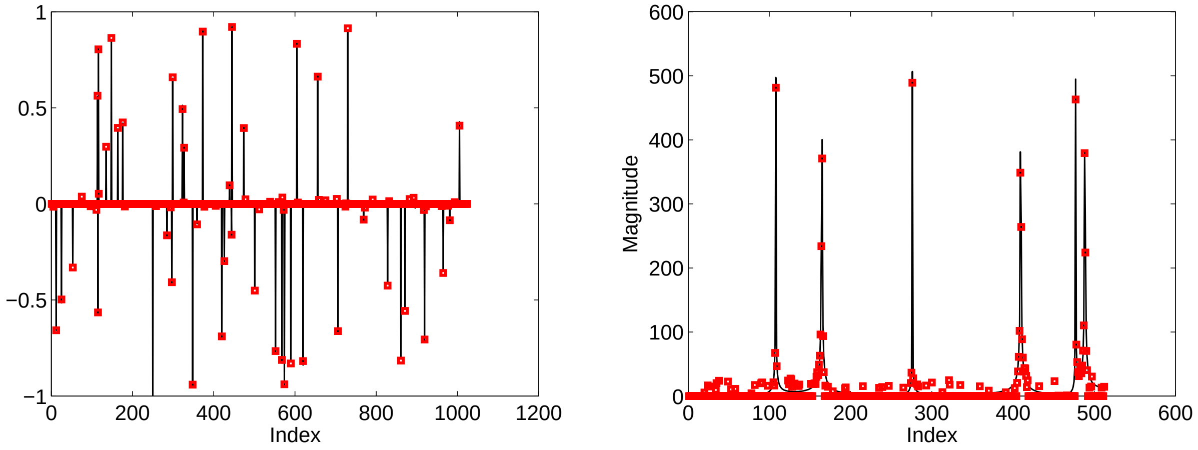
Subgradient Projection Methods for Compressed Sensing



(a) Convergence Time

(b) Ideal normalized error

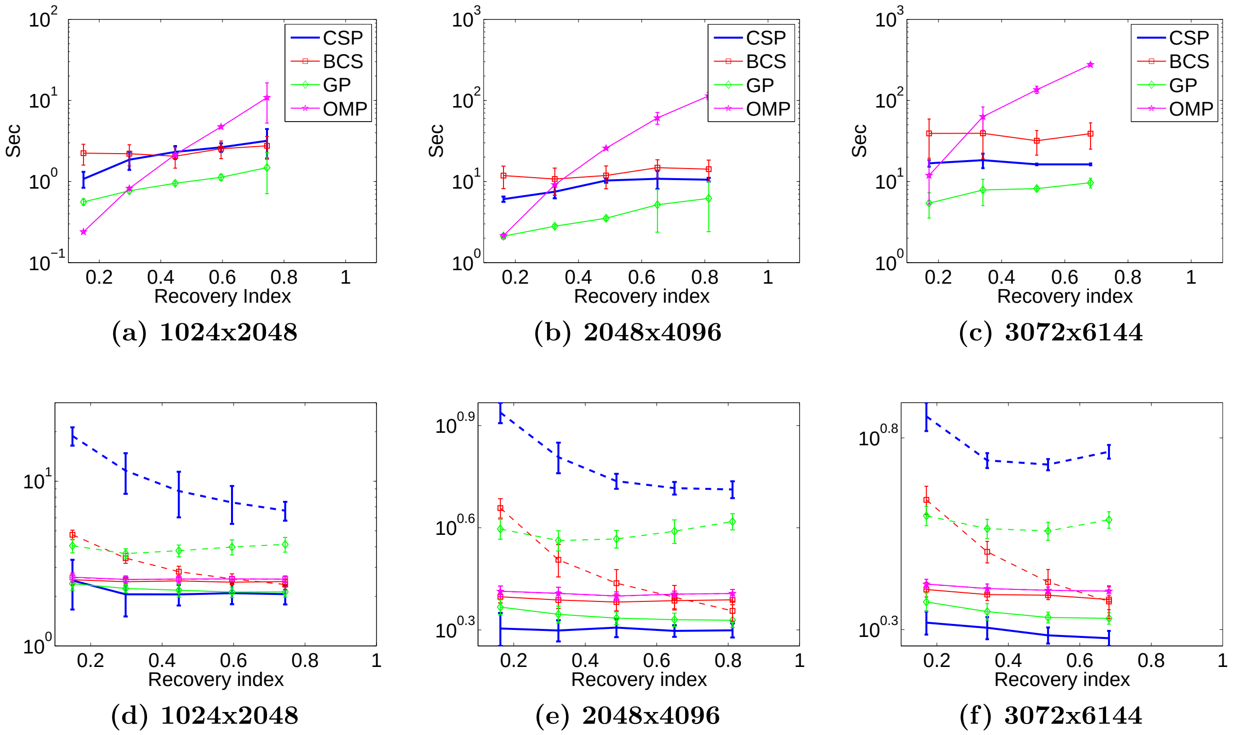
Figure 1. The average performance (over 50 Monte Carlo runs) of the CS methods (including the computationally excessive ones, the LARS and the BP) for problem dimension 512x1024. The performance of the LS augmented versions of the algorithms is depicted via solid lines.



(a) Sparse signal

(b) DFT (compressible) signal

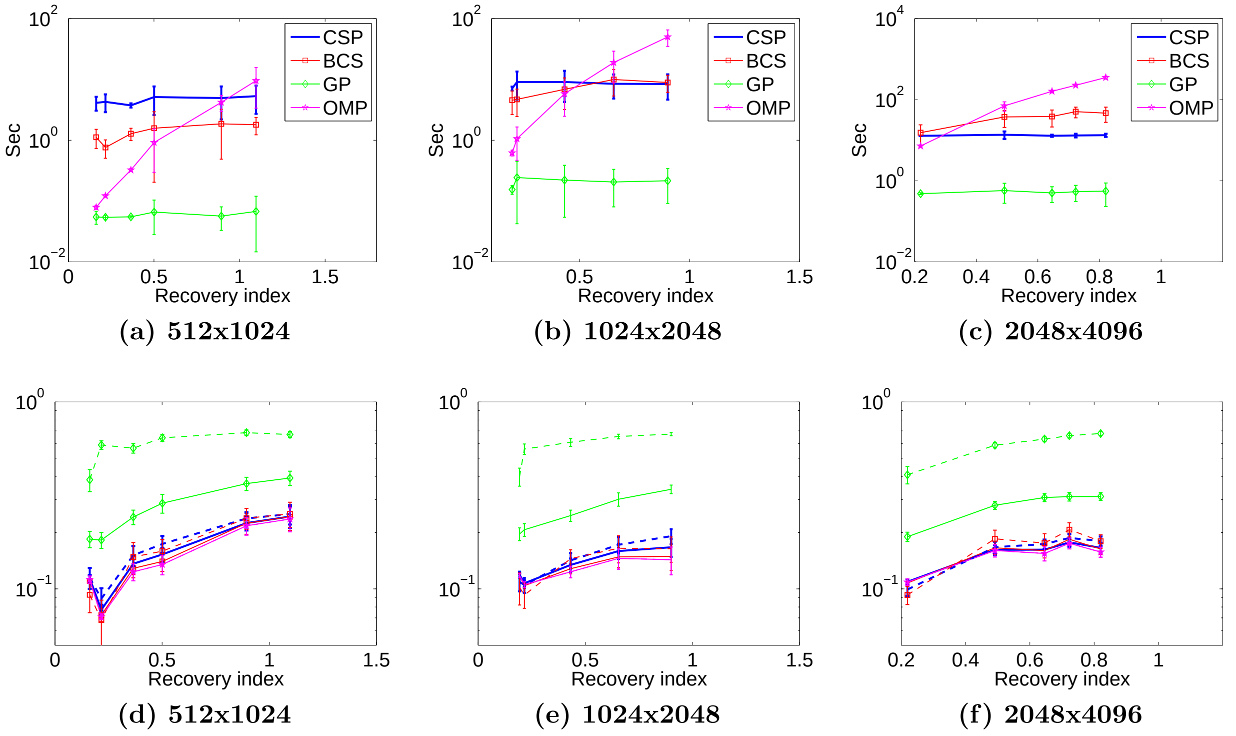
Figure 2. Illustration of the recovery performance of the CSP method for sparse and compressible signals.



(a) 1024x2048 (b) 2048x4096 (c) 3072x6144

(d) 1024x2048 (e) 2048x4096 (f) 3072x6144

Figure 3. The average performance of the CS algorithms for various problem dimensions and sparseness degrees. Showing the convergence time (upper panel) and the *ideal* normalized error (bottom panel). The computationally excessive methods (LARS and BP) are not shown here.



(a) 512x1024 (b) 1024x2048 (c) 2048x4096

(d) 512x1024 (e) 1024x2048 (f) 2048x4096

Figure 4. Recovering a DFT from undersampled data. Showing the average performance of the CS algorithms for various problem dimensions and *effective* sparseness degrees. The convergence time and the normalized error are depicted in the upper and lower panels, respectively.

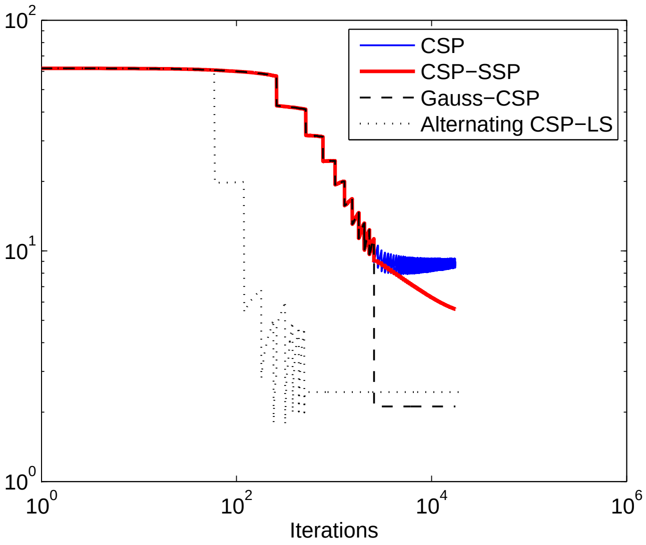
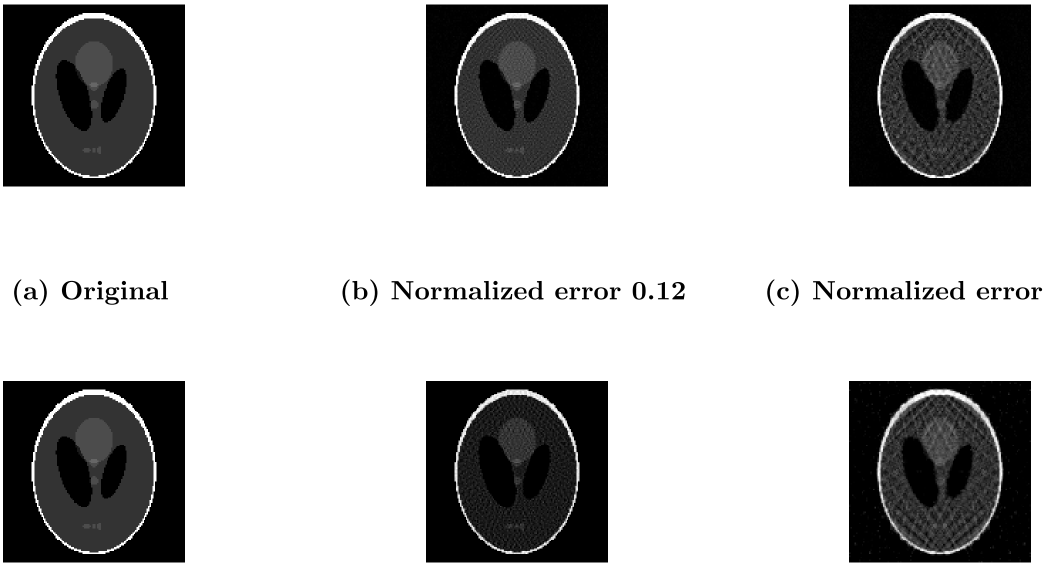


Figure 5. The *ideal* normalized error of the CSP variants . Single run, problem dimension 2048x4096.



(a) Original

(b) Normalized error 0.12

(c) Normalized error 0.22

(d) Original

(e) Normalized error 0.19

(f) Normalized error 0.28

Figure 6. The original and reconstructed 128x128 Shepp-Logan phantom head. Showing the CSP (upper panel) and the GP (lower panel) methods. The middle and right rows depict the recovery error based on 64 and 32 projections, respectively (equivalent to 50% and 25% undersampled data).